REMARKS

35 U.S.C. § 101 Rejection of Claim

In the November 24, 2003 office action, claim 1 is rejected under 35 U.S.C. § 101 because it is directed to non-statutory subject matter. Because the instant application does not have an active claim 1 this rejection was ignored for purposes of this response in accordance with the fax of 7 December 2003 from the Applicant.

35 U.S.C. § 102 Rejection of Claim

In the November 24, 2003 office action, claims 34 - 40, 42 - 48 and 50 - 52 are rejected under 35 U.S.C. § 102 (e) because they are described in U.S. Patent 6,332,163 (hereinafter Bowman-Amuah) that matured from application 09/387,642 filed September 1, 1999). MPEP § 2131 provides that:

> "A claim is anticipated only if each and every element as set forth in the claim is found either expressly or inherently described in a single prior art reference."

The Applicant traverses all U.S.C. § 102 (e) rejections by noting that Bowman Amuah fails to describe elements cited in each of the claims 34 - 40, 42 - 48 and 50 - 52 in the instant application. In some cases Bowman Amuah also teaches away from the method of the instant application. Because the independent claims (34 and 44) are key, a detailed review of the claim 34 is presented below.

Original Claims	34. (amended) A computer readable medium having sequences of instructions stored therein, which when executed cause the processor in a computer to perform a data preparation method, comprising: integrating data from a variety of sources using xml and a common schema to support organization processing.
Background Context provided by Applicant	A computer readable medium for implementing communication services patterns that utilizes a Netcentric Architecture Framework, where the Netcentric Architecture Framework identifies those run-time services required when an application executes in a Netcentric environment. As shown in FIG. 10, the services can be broken down into logical areas: Presentation Services 1000, Information Services 1002,1004, Communication Services 1006,1008, Communication Fabric Services 1010, Transaction Services 1012,1014, Environment Services 1016,1018, Base Services 1020 and Business Logic 1022,1024.

PRESENTATION 1000

Presentation Services enable an application to manage the human-computer interface. This includes capturing user actions and generating resulting events, presenting data to the user, and assisting in the management of the dialog flow of processing. FIG. 13 illustrates several components of the Presentation area of the Netcentric Architecture Framework....

Window System 1300

Typically part of the operating system, the Window System Services provide the base functionality for creating and managing a graphical user interface (GUI)--detecting user actions, managing windows on the display, and displaying information in windows.

Form 1304
Form Services enable applications to use fields to display and collect data. A field may be a traditional 3270-style field used to display or input textual data, or it may be a graphical field such as a check box, a list box or an image. Form Services provide support for:

Display--support the display of various data types (e.g., text, numeric, date, etc.) in various formats (e.g., American/European date, double-byte characters, icons, etc.)

Input/Validation--enable applications to collect information from the user, edit it according to the display options, and perform basic validation such as range or format checks.

Mapping Support--eliminate the need for applications to communicate directly with the windowing system; rather, applications retrieve or display data by automatically copying the contents of a window's fields to a copybook structure in memory. These Services may also be used to automate the merging of application data with pre-defined electronic form templates.

Field Interaction Management—coordinate activity across fields in a window by managing field inter-dependencies and invoking application logic based on the state of fields and user actions. For example, the Field Interaction Manager may disable the "OK" button until all required input fields contain valid data. These services significantly reduce the application logic complexity inherent to an interactive windowed interface....

User Navigation 1306

User Navigation Services provide a user with a way to access or navigate between functions within or across applications. Historically, this has been the role of a text-based menuing system that provides a list of applications or activities for the user to choose

Serial No. 09/940,450   - 15 -

Examiner: Clement Graham
Art Unit: 3628

	from.
	Client/server technologies introduced new navigation metaphors. A method for allowing a user to navigate within an application is to list available functions or information by means of a menu bar with associated pull-down menus or context-sensitive pop-up menus. This method conserves screen real-estate by hiding functions and options within menus, but for this very reason can be more difficult for first time or infrequent users. This point is important when implementing electronic commerce solutions where the target customer may use the application only once or very infrequently (e.g., purchasing auto insurance).
	Additionally, client/server development tools such as Visual Basic and PowerBuilder do not provide specific services for graphical navigation, but the effect can be recreated by selecting (i.e., clicking on) graphical controls, such as picture controls or iconic push-buttons, programmed to launch a particular window.
	A major advantage of the graphical user interface is the fact that it allows multiple windows to be open at one time.
	Web Browser 1308
	Web Browser Services allow users to view and interact with applications and documents made up of varying data types, such as text, graphics, and audio. These services also provide support for navigation within and across documents no matter where they are located, through the use of links embedded into the document content....
	XML: X marks the spot
	HTML 4.0 and Dynamic HTML have given Web authors more control over the ways in which a Web page is displayed. But they have done little to address a growing problem in the developer community: how to access and manage data in Web documents so as to gain more control over document structure. To this end, leading Internet developers devised Extensible Markup Language (XML), a watered-down version of SGML that reduces its complexity while maintaining its flexibility. Like SGML, XML is a meta-language that allows authors to create their own customized tags to identify different types of data on their Web pages.
Prior art cited by the examiner (Column 41 and 42, lines 5 - 30)	In addition to improving document structure, these tags will make it possible to more effectively index and search for information in databases and on the Web.
	XML documents consist of two parts. The first is the document itself, which contains XML tags for identifying data elements and resembles an HTML document. The second part is a DTD that defines the document structure by explaining what the tags mean and how they should be interpreted. In order to view XML

	documents, Web browsers and search engines will need special XML processors called "parsers." Currently, Microsoft's Internet Explorer 4.0 contains two XML parsers: a high-performance parser written in C++ and another one written in Java.
	A number of vendors plan to use XML as the underlying language for new Web standards and applications. Microsoft uses XML for its Channel Definition Format, a Web-based "push" content delivery system included in Internet Explorer 4.0. Netscape will use XML in its Meta Content Framework to describe and store metadata, or collections of information, in forthcoming versions of Communicator. XML is currently playing an important role the realm of electronic commerce via the Open Financial Exchange, an application developed by Microsoft, Intuit, and CheckFree for conducting electronic financial transactions. Similarly, HL7, a healthcare information systems standards organization, is using XML to support electronic data interchange EDI of clinical, financial, and administrative information (http://www.mcis.duke.edu/standards/HL7/sigs/sgml/index.html).
	The future: give us a big SMIL
	The Web has come a long way since the codification of HTML 1.0. It has moved from simple text-based documents that included headings, bulleted lists, and hyperlinks to dynamic pages that support rich graphic images and virtual reality. So what next for the Web? The answer resides in a Synchronized Multimedia Integration Language (SMIL), a new markup language being developed by the W3C. SMIL will allow Web authors to deliver television-like content over the Web using less bandwidth and a simple text editor, rather than intricate scripting.
	SMIL is based on XML and does not represent a specific media format. Instead, SMIL defines the tags that link different media types together. The language enables Web authors to sort multimedia content into separate audio, video, text, and image files and streams which are sent to a user's browser. The SMIL tags then specify the "schedule" for displaying those components by determining whether they should be played together or sequentially. This enables elaborate multimedia presentations to be created out of smaller, less bandwidth-consuming components.
	Implementation Considerations
	Many features such as graphics, frames, etc. supported by Web Browsers today were not available in initial releases. Furthermore, with every new release the functionality supported by Web Browsers keeps growing at a remarkable pace.
Prior art cited by the examiner (column 52, line 45)	Security 1410 The security component prevents unauthorized users from accessing corporate data/resources by providing the users with

	access codes--password & ID--that allows the user to login to the system or execute any (or a particular) application.
Prior art cited by the examiner (column 52, line 45 and column 281, lines 50-55)	Data Handler FIG. 158 illustrates a flowchart for a method 15800 for controlling data. A data retrieval mechanism is provided in operation 15802 for retrieving data from a database. The data retrieval mechanism also writes data to the database. In operation 15804, the data retrieval mechanism is encapsulated in a data handler. A request from a domain object is received for a retrieval of a portion of the data in the database in operation 15806. The data retrieval mechanism is utilized in operation 15808 to retrieve the portion of the data from the database. The portion of the data is passed through the data handler to the domain object in operation 15810. Business objects delegate their data retrieval mechanism to an appropriate handler. This Data Handler can be either be generic or specific to each type of domain object used. To minimize the impact of changes, dependencies on the database schema or data retrieval mechanism within the handler could be managed via code generation. In this manner, the physical data access is separated from pure business logic.
Summary of relevant portion of Bowman Amuah	A computer readable medium for implementing communication services patterns that utilizes a Netcentric Architecture Framework, where the Netcentric Architecture Framework identifies those run-time services required when an application executes in a Netcentric environment. The services can be broken down into logical areas: Presentation Services 1000, Information Services 1002,1004, Communication Services 1006,1008, Communication Fabric Services 1010, Transaction Services 1012,1014, Environment Services 1016,1018, Base Services 1020 and Business Logic 1022,1024. The presentation services in Bowman Amuah support the use of XML to more effectively index and search for information in databases and on the Web. The information services in Bowman Amuah support the use of a data handler that can minimize dependencies on the database schema via code generation.
Other information	Independent applications plan to use XML more and more and some already use it to support electronic transaction processing
What was not described	1) the integration of data, 2) the use of a common schema – Bowman Amuah teaches away by using a different approach – schema dependencies can be/are handled via code generation
Conclusion	Cited references do not describe or anticipate all elements of claim 34, in fact, Bowman Amuah teaches a different approach

The Table below summarizes the traversal of the § 102 rejections by claim. The Applicant's amendment/reply of 19 August 2003 provides detailed support.

Serial No. 09/940,450   - 18 -   Examiner: Clement Graham
                                                                    Art Unit: 3628

Original claims	Traversal includes:
Claims 34 & 35	Cited references do not describe or anticipate all elements of claim 34 or claim 35, teaches away from method of instant application
Claim 36	Cited references do not describe or anticipate any elements of claim 36.
Claim 37	Cited references do not describe or anticipate any elements of claim 37.
Claim 38	Cited references do not describe or anticipate any elements of claim 38
Claim 39	Cited references do not describe or anticipate any elements of claim 39
Claim 40	Cited references do not describe or anticipate any elements of claim 40
Claim 42	Cited references do not describe or anticipate an element of claim 42
Claim 43	Cited references do not describe or anticipate an element of claim 43, teaches away from method of instant application
Claim 44	Cited references do not describe or anticipate elements of claim 44
Claim 45	Cited references do not describe or anticipate any elements of claim 45, teaches away from method of instant application
Claim 46	Cited references do not describe or anticipate any elements of claim 46.
Claim 47	Cited references do not describe or anticipate any elements of claim 47
Claim 48	Cited references do not describe or anticipate any elements of claim 48.
Claim 50	Cited references do not describe or anticipate an element of claim 50
Claim 51	Cited references do not describe or anticipate an element of claim 51
Claim 52	Cited references do not describe or anticipate an element of claim 52, teaches away from method of instant application

35 U.S.C. § 103 Rejection of Claim

In the November 24, 2003 office action, claims 41 and 49 are rejected under 35 U.S.C. § 103 because they would be obvious given the prior art. The prior art cited by the Examiner was Bowman-Amuah and the Examiner's stated opinion that receiving

Serial No. 09/940,450   - 19 -

Examiner: Clement Graham
Art Unit: 3628

data related to different groups of systems is old and well known in the art of the transmission of data. The Examiner's opinion is not relevant to the instant application which deals with the integration of data to support organization processing. As detailed above, Bowman-Amuah fails to describe one or more elements contained in every claim and in many cases it teaches away from the method described in the instant application. Combining these last two statements the Applicant traverses this rejection by noting that the Examiner has not established the prima facie case for obviousness that is required to substantiate a rejection under 35 U.S.C. § 103.

In the November 24, 2003 office action, claims 53 - 60 are similarly rejected under 35 U.S.C. § 103 because they would be obvious given the prior art. The prior art cited by the Examiner was Bowman-Amuah and the Examiner's statement that transforming data from a variety of systems into a probabilistic model that quantifies the value contribution of elements of value to a value of an enterprise by category; capturing proposed changes in element value drivers, using the element impact model to simulate the impact of the proposed changes on enterprise financial performance; and displaying the result of the simulation using a paper document or electronic display is old and well known in the art. The Examiner also repeats one of the apparent errors noted previously by claiming that Bowman-Amuah describes the different types of organizations and expresses unsupported opinions regarding elements of value, predictive models, categories of value and the use of quantified element impacts for determining element values. As detailed above, Bowman-Amuah fails to describe one or more elements contained in every claim and in many cases it teaches away from the method described in the instant application. The Applicant traverses the 35 U.S.C. § 103 rejections by noting that as a result the Examiner has failed to establish the prima facie case for obviousness that is required to substantiate the rejections under 35 U.S.C. § 103.

In the November 24, 2003 office action, claim 1-11 are rejected under 35 U.S.C. § 103 because it would be obvious to modify Bowman-Amuah to use the teachings of Subramanian et al (US Patent 6,546,381) to create one or more tools for organization management (claim 62?). Because the instant application does not have active claims 1-11, this rejection was ignored for purposes of this response in accordance with the fax of 7 December 2003 from the Applicant.

In the November 24, 2003 office action, claims 61-134 were not rejected. However, the Examiner states that it would be obvious to modify Bowman-Amuah to use the teachings of Subramanian et al (US Patent 6,546,381) to create one or more tools for organization management. The Examiner quotes the claim incorrectly and states that Bowman-Amuah discloses a computer readable medium having sequences of instructions stored therein, which when executed cause the processors in a plurality of computers that have been connected via a network to perform a data preparation method, comprising integrating data from a variety of sources in accordance with a common schema. The Applicant notes that Bowman-Amuah does not teach the integration of data in accordance with a common schema, as described previously, so the comments concerning a combination are moot.